Supplementary Material to: Cross-Company collaboration to leverage analytics for clinical quality and accelerate drug development - the IMPALA industry group (Ménard et. al.)

Contents

1 Load

```
suppressPackageStartupMessages( library(tidyverse) )
suppressPackageStartupMessages( library(knitr) )
suppressPackageStartupMessages( library(simaerep) )
```

2 Introduction

Simulate adverse event reporting in clinical trials with the goal of detecting under-reporting sites.

Monitoring of Adverse Event (AE) reporting in clinical trials is important for patient safety. We use bootstrap-based simulation to assign an AE under-reporting probability to each site in a clinical trial. The method is inspired by the 'infer' R package and Allen Downey's blog article: "There is only one test!".

2.1 Publication

Koneswarakantha, B., Barmaz, Y., Ménard, T. et al. Follow-up on the Use of Advanced Analytics for Clinical Quality Assurance: Bootstrap Resampling to Enhance Detection of Adverse Event Under-Reporting. Drug Saf (2020). https://doi.org/10.1007/s40264-020-01011-5

2.2 Adverse Events

An adverse event (AE) is any untoward medical occurrence in a patient or clinical investigation subject administered a pharmaceutical product and which does not necessarily have a causal relationship with this treatment. It is important for patient safety that AEs are reported back to the sponsor. An important part of quality monitoring is to detect clinical trial sites that are not propagating all of the AEs reported to them by their patients to the sponsor. In a clinical trial patients follow a strict visiting schedule at which times treatments are given and exams are being performed. Typically AEs get reported by a patient when they are at an on-site visit at the clinic. So the total number of AEs reported by a site depends on the number of patients enrolled at the site and the total number of visits.

2.3 Algorithm

In a nutshell we will perform the following steps.

1) Record visit_med75, number of patients that have reached visit_med75, mean cumulative AE count at visit_med75 for a clinical trial site.
2) Create patient pool with all patients in study that have reached visit_med75 as determined in 1) and their cumulative AE count at visit_med75.
3) Draw with replacement as many patients from 2) as determined in 1) and calculate the mean cumulative AE count of the draw (figure 1b).
4) Repeat 3) 1000 times.
5) Calculate probability of obtaining mean cumulative AE count or lower as obtained in 1) based on results of 4).
6) Repeat 1-5) for all sites in trial.
7) Adjust probabilities using the Benjamin Hochberg Procedure in order to correct alpha error.

2.4 Sample Data

2.4.1 Patient

Patient level AE data is characterized by the number of consecutive visits and the number of AEs that have been reported each time. For the maximum consecutive visit we sample from a normal distribution and for the AEs reported at each visit we sample from a poisson distribution.

Here we simulate the AEs generated by 3 patients

```
set.seed(1)

replicate(
   3,
   sim_test_data_patient(
    .f_sample_max_visit = function() rnorm(1, mean = 20, sd = 4),
    .f_sample_ae_per_visit = function(max_visit) rpois(max_visit, 0.5)
  )
)
```

```
## [[1]]
##  [1]  0  1  1  2  4  5  6  6  6  6  7  7  8  8  9 12 12
##
## [[2]]
##  [1] 0 1 1 1 1 1 1 2 2 2 2 2 2 3 4 5 5 6 6 7 8 9 9
##
## [[3]]
##  [1] 0 0 1 2 2 3 3 3 3 3 3 3 4 4 6 6 6 6 7 7
```

2.4.2 Study

In order to simulate patient data for an entire study we assume make the simplification that all sites have the same number of patients. Further we specify a fraction of sites that is under-reporting AEs

```
df_visit <- sim_test_data_study(
  n_pat = 120,
  n_sites = 6,
  frac_site_with_ur = 0.4,
  ur_rate = 0.6,
  max_visit_mean = 20,
  max_visit_sd = 4,
  ae_per_visit_mean = 0.5
)

df_visit %>%
  head(10) %>%
  kable()
```

patnum	site_number	is_ur	max_visit_mean	max_visit_sd	ae_per_visit_mean	visit	n_ae
P000001	S0001	TRUE	20	4	0.2	1	0
P000001	S0001	TRUE	20	4	0.2	2	1
P000001	S0001	TRUE	20	4	0.2	3	1
P000001	S0001	TRUE	20	4	0.2	4	2
P000001	S0001	TRUE	20	4	0.2	5	2
P000001	S0001	TRUE	20	4	0.2	6	2
P000001	S0001	TRUE	20	4	0.2	7	2
P000001	S0001	TRUE	20	4	0.2	8	3
P000001	S0001	TRUE	20	4	0.2	9	4
P000001	S0001	TRUE	20	4	0.2	10	4

```
df_visit %>%
  select(site_number, is_ur) %>%
  distinct() %>%
  kable()
```

site_number	is_ur
S0001	TRUE
S0002	TRUE
S0003	FALSE
S0004	FALSE
S0005	FALSE
S0006	FALSE

In our sample data 2 sites (S0001 and S0002) are under-reporting AEs

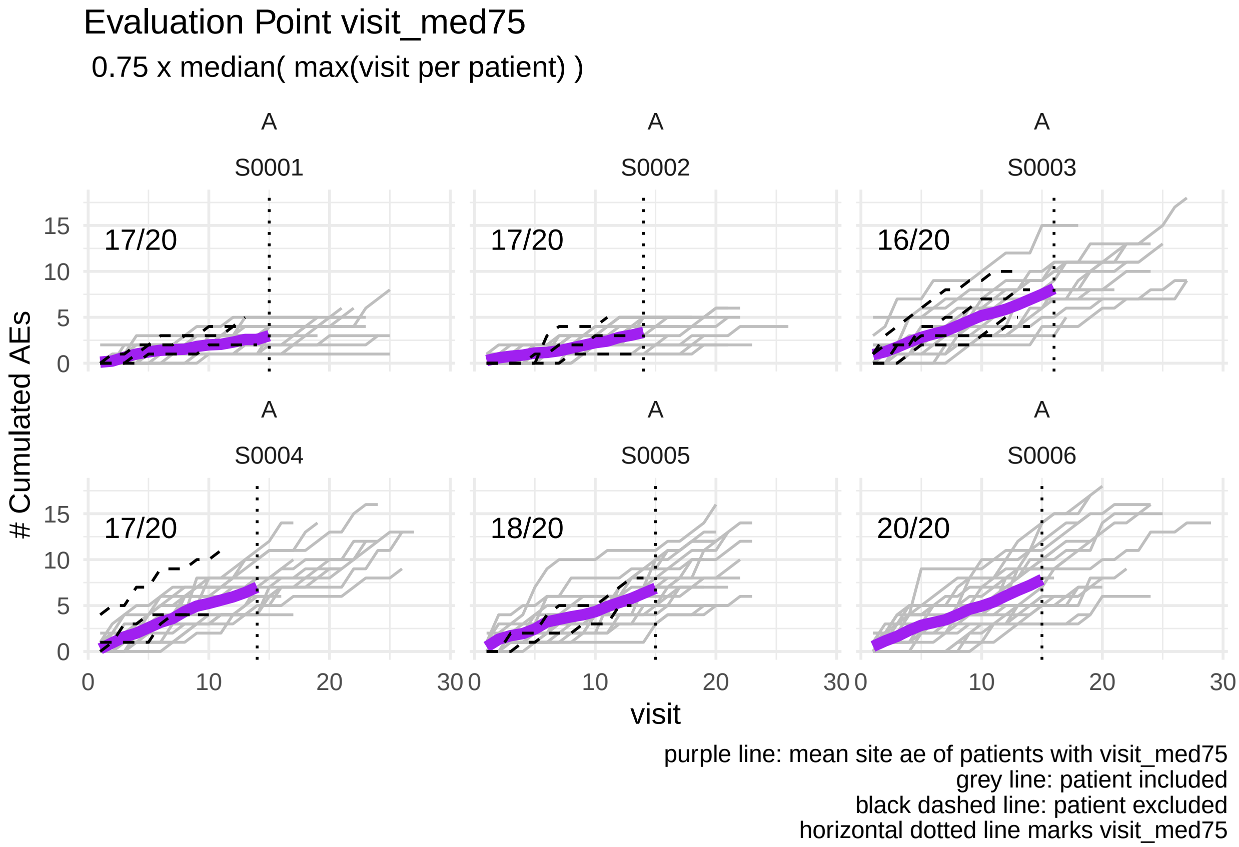
2.5 Specifying the Evaluation Point visit_med75

In an ongoing trial all patients will have a different number of consecutive visits. To find a cut-off visit to normalize the data we specify a single evaluation point for a given site based on the number of visits of the patient population. We take the median of the maximum visit of all the patients and multiply by 0.75 (visit_med75). For this we use `site_aggr` to aggregate the simulated visit level data to site level.

```
df_visit$study_id <- "A"
df_site <- site_aggr(df_visit)

df_site %>%
  kable()
```

study_id	site_number	n_patients	visit_med75	n_pat_with_med75	mean_ae_site_med75
A	S0001	20	15	17	3.000000
A	S0002	20	14	17	3.352941
A	S0003	20	16	16	8.125000
A	S0004	20	14	17	7.000000
A	S0005	20	15	18	6.888889
A	S0006	20	15	20	7.850000

```
plot_visit_med75(df_visit, df_site, study_id_str = "A", n_site = 6)
```



Using the visit_med75 will drop a few patients (dashed lines) but will by definition at least include 50% of all patients favoring patients that have higher number of visits. By looking at the mean ae development (purple line) we can already easily spot the two under-reporting sites. Same goes for looking at the `mean_ae_site_med75` column in `df_site`. Next we will use bootstrap simulations to determine the probability for obtaining each mean ae value or lower at visit_med75 by chance.

2.6 Bootstrap Simulations

2.6.1 Advantage Over Classic Statistical Tests

We could use a classical parametric test and calculate the probability with which we can reject the NULL hypothesis for the AE counts that we observe at the visit_med75.

As we sample the AEs from a poisson distribution the R implementation of the poisson.test would be appropriate.

But there are four major problems that we often encounter when we try to describe real life count data with the poisson distribution which is only described by a single parameter:

- over and underdispersion
- skewness (long right tail)
- inflated zeros
- variance mean relationship might not be fixed

(see Distribution for Modelling Location Scale and Shape 5.1.2)

The true distribution of the AE counts will vary from study to study with the degree of the influence of these 4 problems being unknown unless thoroughly investigated.

With the non-parametric approach that we propose we do not need to worry about these statistical assumptions. Since the distribution of the AE count from the simulated patient pool we draw from will be very close to the unknown true distribution of the underlying AE generating process.

2.6.2 Disadvantages Over Classic Statistical Tests

- **Upper Limit for Uncompliant Sites**. We simulate an underlying compliant patient population using the data we are given. If the fraction of uncompliant examples becomes too high, detection rates will decrease. We find that detection rates start decreasing with 30-50% of under-reporting sites and are not usable anymore if a majority $> 50\%$ of all sites are under-reporting see article on usability limits.
- **Lower Probability Limit** The number of repeats determines the smallest probability greater than zero, for example a for r=1000 the smallest value greater than zero is 0.001
- **Computationally Expensive**

2.6.3 Methodology

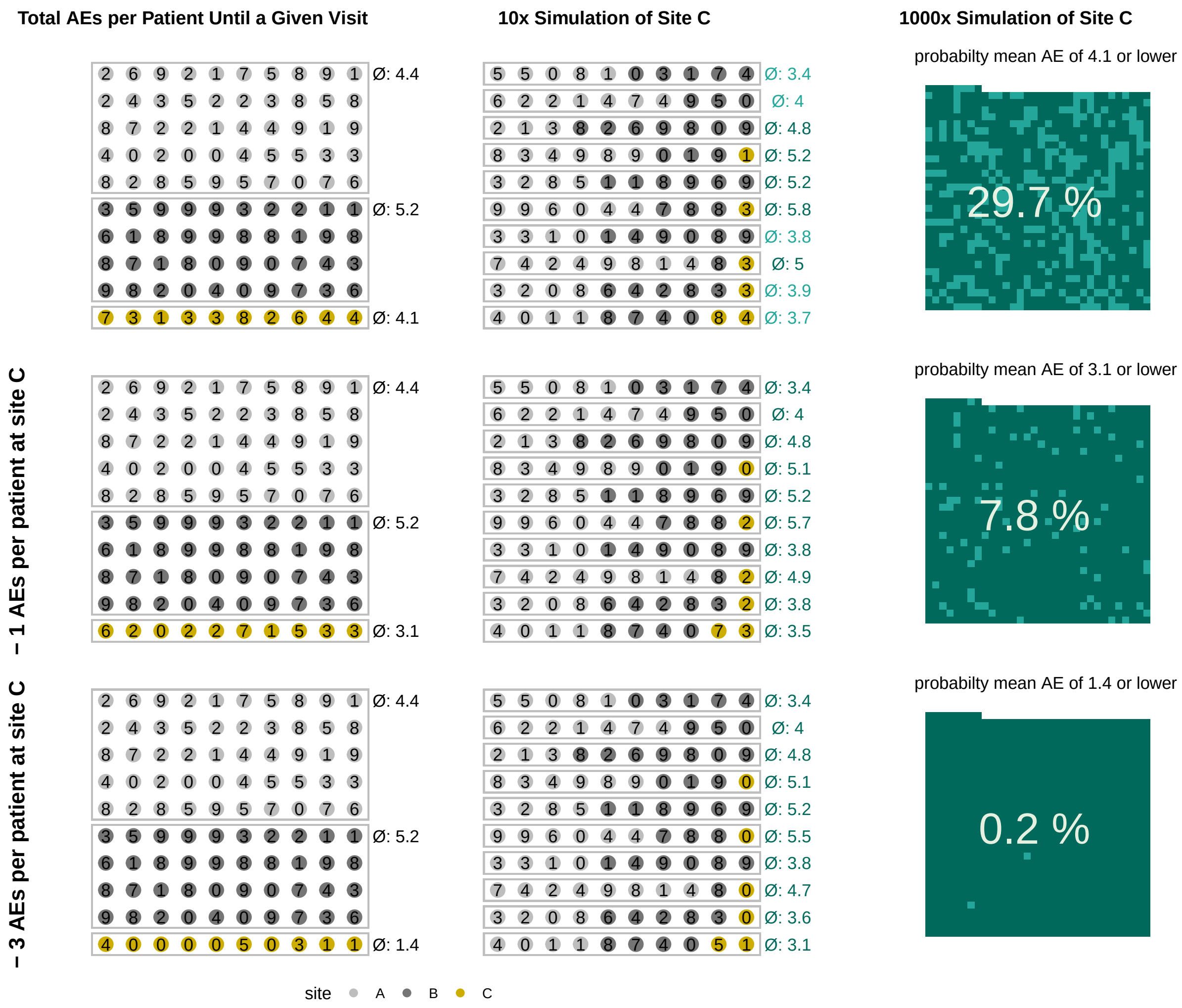
How likely is it to get a mean AE value that is equal or lower to what we observe at site C with the same number of patients?

For illustration purposes we start to simulate 10 hypothetical patient groups for site C by drawing (with replacement) from all patients in the study marking those for which we have obtained an equal or lower mean AE value than initially observed (middle).

Instead of 10 times we simulate 1000 times and count how many times we have observed an equal or lower mean AE value than initially observed and convert it to a percentage (right).

To illustrate the effect of under-reporting we repeat the same process after having removed 2 AEs per patient from site C. We see how the probability for obtaining an equal or lower mean AE value than initially observed decreases (bottom).

```r
plot_sim_examples(size_dot = 4, size_raster_label = 10)
```



2.6.4 Application

We can run the above described simulation and as a benchmark also perform a poisson test.

```r
df_sim_sites <- sim_sites(df_site, df_visit, r = 1000, poisson_test = TRUE)

df_sim_sites %>%
  kable()
```

study_id	site_number	visit_med75	mean_ae_site_med75	mean_ae_study_med75	pval	prob_low
A	S0001	15	3.000000	6.712644	0.0e+00	0
A	S0002	14	3.352941	6.021739	7.5e-06	0
A	S0003	16	8.125000	6.270588	1.0e+00	1
A	S0004	14	7.000000	5.347826	1.0e+00	1
A	S0005	15	6.888889	5.941860	1.0e+00	1
A	S0006	15	7.850000	5.690476	1.0e+00	1

We find that the probability of getting a mean ae at visit_med75 for sites S0001 and S0002 is 0 or near zero.

2.7 Alpha Error Correction

Our simulated test data set consists of just 6 sites. However, it is not uncommon for 100 sites or more to participate in the same clinical trial. This would mean that we need to perform 100 statistical tests, and applying a 5% significance threshold would lead to on average of 5 False Positives (FP). We therefore need to adjust the calculated p-values and bootstrapped probabilities using `stats::p.adjust(p, method = "BH")`, which applies the Benjamin Hochberg Procedure. `eval_sites()` uses the the inverted adjusted values to calculates the final bootstrapped AE under-reporting probability (prob_low_prob_ur) and includes the poisson test derived under-reporting probability as a reference (pval_prob_ur).

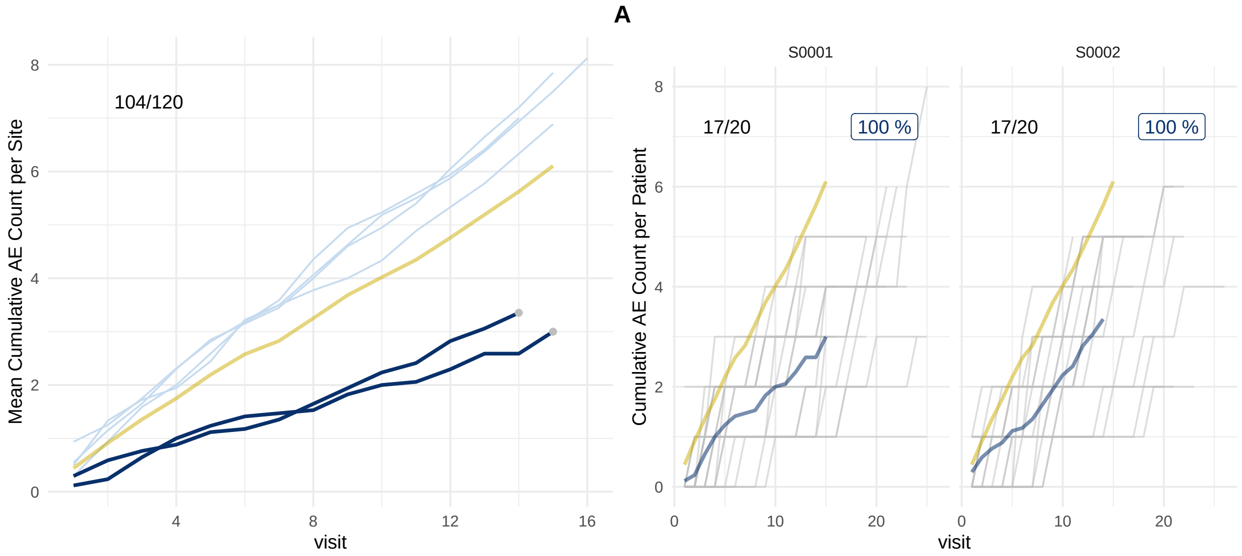
```
df_eval <- eval_sites(df_sim_sites, method = "BH")

df_eval %>%
  select(- visit_med75, - mean_ae_site_med75, - mean_ae_study_med75) %>%
  kable()
```

study_id	site_number	pval	prob_low	pval_adj	pval_prob_ur	prob_low_adj	prob_low_prob_ur
A	S0001	0.0e+00	0	0.00e+00	1.0000000	0	1
A	S0002	7.5e-06	0	2.24e-05	0.9999776	0	1
A	S0003	1.0e+00	1	1.00e+00	0.0000000	1	0
A	S0004	1.0e+00	1	1.00e+00	0.0000000	1	0
A	S0005	1.0e+00	1	1.00e+00	0.0000000	1	0
A	S0006	1.0e+00	1	1.00e+00	0.0000000	1	0

2.8 Plot Results

`plot_study` will plot mean ae development of all sites and visit level data for each flagged site with an AE under-reporting threshold of 95%.

```
plot_study(df_visit, df_site, df_eval, study = "A")
```


A
104/120
Mean Cumulative AE Count per Site
visit
S0001
S0002
17/20
17/20
Cumulative AE Count per Patient
visit